SYSTEM AND METHOD FOR CLOCK SIGNAL SYNCHRONIZATION

Field of the Invention

The present invention relates, in general, to a system and method for signal synchronization and, more particularly, to a system and a method for locking the clock signal of an oscillator to a data stream in data communications.

Background of the Invention

Conventional data communication circuits require a precision timing component to provide a reference frequency clock signal to an external device that is coupled via a signal transmission bus to a host. A precision timing component in such communication circuits usually includes a crystal oscillation element. An internally based timer tunes the clock signal of the crystal oscillation element to match the clock signal to the incoming data stream from the host. Typically, a phase lock loop (PLL) or a delay lock loop (DLL) in the timer serves the functions of tuning and locking the clock signal through data training, phase shifting, phase selection, or the like. The crystal oscillator is expensive. The internally based timer typically requires long training sequences to tune a PLL or DLL, which may not be available in modern applications, such as universal serial bus (USB) applications.

An alternative approach for locking the clock signal to the incoming data stream includes generating a clock signal from a current-controlled oscillator (ICO) or a voltage-controlled oscillator (VCO), analyzing the rates of an incoming data stream in at least two periods to generate two or more control signals, and adjusting the frequency of the clock signal in response to the control signals. Adjusting the frequency of the clock signal operates in an analog mode and generally includes at least two steps: a coarse tuning step followed by a fine tuning step. The ICO or VCO is an application specific integrated circuit (ASIC) that takes a large chip area and thereby increases the cost of the communication circuit. The analog multiple step tuning process is slow and complicated. The performance of the analog tuning circuit is susceptible to process and temperature variations. Complicated processing and circuit schemes may be required to reduce the variations and improve the performance and reliability of the tuning process.

Accordingly, it would be advantageous to have a cost efficient system and a process for synchronizing a clock signal to a data signal. It is desirable for the system to be simple and silicon area efficient. It is also desirable for the synchronization process to be fast

and reliable. It is of further advantages for the system and the process to be unsusceptible to variations in chip fabrication processes and operation conditions.

Brief Description of the Drawings

Figure 1 is a block diagram illustrating a clock signal synchronization system in accordance with the present invention;

Figure 2 is a timing diagram illustrating a token packet in a universal serial bus communication protocol in accordance with the present invention;

Figure 3 is a flow diagram illustrating a process for digitally analyzing a packet in accordance with the present invention; and

Figure 4 is a flow diagram illustrating a process for digitally synchronizing a clock signal to a packet in accordance with the present invention.

Detailed Description of various Embodiments

Various embodiments of the present invention are described herein below with reference to the figures, in which elements of similar structures or functions are represented by like reference numerals throughout the figures. It should be noted that the figures are only intended to facilitate the description of the preferred embodiments of the present invention. They are not intended as an exhaustive description of the present invention or as a limitation on the scope of the present invention.

Figure 1 is a block diagram illustrating a precision timing component or a clock signal synchronization system 101 in accordance with the present invention. By way of example, Fig. 1 shows system 101 is a part of a universal serial bus (USB) device 100 and functions for generate a clock signal synchronized with a packet received from a host (not shown in Fig. 1) via a USB bus 110. In Fig. 1 an element 102 represents portions of USB device 100 other than clock signal synchronization system 101. Element 102, which also referred to as a data processing element, may include a USB control circuit and other components of USB device 100. The USB control circuit, which is sometimes also referred to as a USB driver, functions to control the data transfer between the host and an external or slave device, e.g., USB device 100, via USB bus 110.

USB device 100 can be any kind of devices that communicates with a host via USB bus 110. Examples of USB device 100 include, but are not limited to, USB mouse for moving a cursor on the host computer screen and making commands to the host computer,

USB memory device (e.g., USB hard drive, USB CD-ROM, USB rewritable CD, USB rewritable DVD, USB flash memory, etc.), USB multimedia devices (e.g., USB CD player, USB DVD player, USB MP3 player, etc.). USB bus 110 is coupled between USB device 100 and the host or master device. As well known in the art, USB bus 110 includes four wires or lines, two of which are data transmission (D+) line 112 and complementary data transmission (D-) line 114, and the other two are power supply line 115, and ground line 117. In accordance with an embodiment of the present invention, clock signal synchronization system 101 is fabricated on an integrated circuit chip that realizes part or whole functions of USB device 100.

Clock signal synchronization system 101 includes an oscillator 103 serving as a reference signal generator, a data sequence analyzer 104, and a synchronized clock signal generator 105. In accordance with the present invention, oscillator 103 provides a reference frequency signal to data sequence analyzer 104 and to synchronized clock signal generator 105. Data sequence analyzer 104 identifies and analyzes an incoming data stream and generates a digital control signal. In response to the digital control signal from data sequence analyzer 104 and the reference frequency signal from oscillator 103, clock signal generator 105 generates a clock signal that is synchronized or locked to the incoming data stream. In a specific embodiment, signal generator 105 includes counters 106 and 108 as shown in Fig. 1. The operations of data signal sequence analyzer 104 and synchronized clock signal generator 105 in accordance with a preferred embodiment of the present invention are described herein after with Figs. 3 and 4.

In accordance with a preferred embodiment of the present invention, oscillator 103 is a resistor-capacitor (RC) oscillator generating a fixed frequency signal. Compared with other types of oscillation circuits such as crystal oscillator, ICO, VCO, etc., RC oscillator 103 is simple and inexpensive. RC oscillator 103 also has a small foot print, i.e., it is very silicon area efficient. It should be noted that, although oscillator 103 is described herein as an RC oscillator, this is not intended as a limitation on the scope of the present invention. In accordance with the present invention, other types of clock source, e.g., clock on another chip, crystal oscillator, ceramic oscillator, ICO, VCO, etc., can also serve as oscillator 103 in system 101.

Figure 2 is a timing diagram illustrating a token packet 200 in a USB communication protocol in accordance with the present invention. By way of example, Fig. 2

shows the first ten bits of a token packet during a low speed data transmission according to USB Version 1.1 protocol. For a full speed data transmission in the USB Version 1.1 protocol, the voltage levels at the D+ and D- lines are opposite to those shown in Fig. 2. The first eight bits of token packet 200 form a synchronization (sync) field and the last two bits are part of a packet identifier (pid) field of token packet 200.

The first ten bits of token packet 200 at the D+ line is 1010101 110. Figure 2 also shows a wave 210 corresponding to such a digital value. An edge in wave 210 represents a change in the bit value in token packet 200. Wave 210 has rising edges 201, 203, 205, and 207 corresponding to the voltage level at the D+ line changing from low to high or bit value from 0 to 1. Wave 210 also has falling edges 202, 204, 206, and 208 corresponding to the voltage level at the D+ line changing from high to low or bit value from 1 to 0. Before the host transmits token packet 200, USB device 100 is in an idle state with the voltage at the D+ line at a low level corresponding to a bit value of 0 and the voltage at the D- line at a high level corresponding to a bit value of 1. Rising edge 201 in wave 210 indicates the arrival of token packet 200.

Figure 3 is a flow diagram illustrating a process 300 for digitally analyzing a packet in accordance with the present invention. By way of example, data analyzing process 300 can be implemented in data sequence analyzer 104 to generate the digital control signal for clock signal generator 105 shown in Fig. 1.

Referring to Figs. 1, 2, and 3 simultaneously, when first powered up, element 102 transmits a reset signal to data sequence analyzer 104 and synchronized clock signal generator 105. In a USB protocol, the reset signal is represented by setting voltage levels at both D+ and D- lines at low for a predetermined period, e.g., 10 milliseconds (ms). In response to the reset signal, data sequence analyzer 104 performs initialization in a step 301. Upon initialization, data sequence analyzer 104 sets the digital control signal to a predetermined initial value. In accordance with a specific embodiment, digital control signal has eight bits and the predetermined initial value is 128.

In a subsequent step 302, data sequence analyzer 104 detects an end of packet (EOP) signal. In accordance with a specific embodiment of the present invention, the EOP is indicated by the voltage levels at both the D+ and D- lines in a USB bus staying low for a predetermined period, e.g., a period equal to or greater than one bit period. After the EOP signal, the USB bus generally enters an idle state, waiting for the host to send out a packet.

While in the idle state, data sequence analyzer 104 detects an incoming packet in a step 303. In accordance with a preferred embodiment of the present invention, the start of an incoming packet is indicated by a change in the voltage levels at the D+ and D- lines of the USB bus. For example, rising edge 201 in wave 200 (shown in Fig. 2) represents a low to high voltage level change in the D+ line and indicates the incoming packet.

After detecting the incoming packet, data sequence analyzer 104 seeks to identify the type of the packet in a step 304. Specifically, data sequence analyzer 104 verifies whether the incoming packet is a token packet in step 304. In a specific embodiment of the present invention, data sequence analyzer 104 identifies the incoming packet as a token packet in response to the packet satisfying three preset conditions. The first condition is a time duration or interval between the first falling edge (edge 202 in Fig. 2) and the second rising edge (edge 203 in Fig. 2) being approximately equal to a time duration or interval between edge 203 and the second falling edge (edge 204) in wave 210 represents the voltage level at the D+ line. The second condition is a time duration between the first falling edge (edge 202) and the second falling edge (edge 204) being approximately equal to a time duration between edge 204 and the third falling edge (edge 206) in wave 210. The third condition is a time duration between the first falling edge (edge 202) and the third falling edge (edge 206) being approximately equal to a time duration between edge 206 and the fourth falling edge (edge 208) in wave 210. In accordance with the present invention, any timing signal can be used to measure the time durations. For example, in a preferred embodiment of the present invention, the reference frequency signal from RC oscillator 103 is used for the time measurement. Generally speaking, the higher frequency of the reference frequency signal is, the more accurate the time measurement will be. In accordance with a preferred embodiment, two time durations are considered to be approximately equal to each other if a difference there between is less than about ten percent (10%). In accordance with another preferred embodiment, two time durations are considered to be approximately equal to each other if a difference there between is less than about five percent (5%). Other criteria are within the spirit of the present invention and also fall into the scope of the present invention.

In accordance with an embodiment of the present invention, the reference frequency signal from RC oscillator 103 is used for measuring time and verifying the conditions in process 300. It should be understood that process 300 is not limited to identifying the incoming packet using the conditions described herein with reference to

step 304. Other schemes can also be used to identify the incoming packet. Preferably, packet identification does not rely on the first edge, e.g., edge 201 in Fig. 2, of the wave corresponding to a packet because the first edge of the packet is often unstable.

In response to the incoming packet not being a token packet, process 300 returns to step 303 and waits for a subsequent incoming packet. If the incoming packet is identified as a token packet, process 300 proceeds to a step 305. In step 305, process 300 assigns a value to the digital control signal. In accordance with a specific embodiment of the present invention, process 300 assigns a value equal to the number of periods of the reference frequency signal generated by RC oscillator 103 in a time duration of interval between the first falling edge (edge 202) and the fourth falling edge (edge 208) in wave 210 of the token packet. This time interval is equal to eight times of the bit period of the token packet. Specifically, this time interval covers a time duration from the beginning of the second bit to the beginning of the tenth bit in token packet 200. In a clock signal synchronization process 400 described herein after with reference to Fig. 4, the assigned value is used to generate a clock signal synchronized with the incoming packet. Depending on how the digital control signal is used in generating the synchronized clock signal, data analyzing process 300 may assign different values to the digital control signal in step 305. The assigned value preferably represents a relationship between the data rate of the incoming packet and the reference frequency signal. In addition, the assigned value preferably does not depend on the time of the first edge, e.g., edge 201 in wave 210, because it may be unstable.

After assigning the value to the digital control signal, process 300 returns to step 302 and waits for a new incoming packet. In response to the new incoming packet, process 300 repeats steps 303, 304, and 305 to identifying the packet and assign a value to the digital control signal in response to the packet being a token packet. In accordance with a preferred embodiment of the presentation, the digital control signal is used to synchronize or lock a clock signal to a data stream.

Figure 4 is a flow diagram illustrating a process 400 for digitally synchronizing a clock signal to a packet in accordance with the present invention. By way of example, process 400 can be implemented in synchronized clock signal generator 105 to generate a clock signal locked to a data stream transmitted from the host via USB bus 110 shown in Fig. 1. In accordance with an embodiment of the present invention, process 400 digitally generates a clock signal synchronized with the a packet in the data stream by using the digital

control signal of data sequence analyzer 104 to calculate the number of cycles of the reference frequency signal of RC oscillator 103. In a preferred embodiment of the present invention, process 400 is activated after power on. Upon activation, in a step 402, counters 106 and 108 in synchronized clock signal generator 105 (shown in Fig. 1) are initialized and set to zero in response to the reset signal from element 102. After initialization, process 400 is repeatedly performed from a step 403 of detecting a bit value change, as shown in Fig. 4 and described herein after. In a preferred embodiment, process 400 has a cycle time equal to the period of the reference frequency signal generated at RC oscillator 103. A higher frequency for the reference frequency signal results in more cycles per unit time and more accurate synchronization.

At the beginning of each cycle of the reference signal of RC oscillator 103, process 400, in step 403, checks the signal level at element 102 to see whether USB device 100 is receiving or waiting for a packet from the host. IfUSB device 100 is receiving or waiting for packets for the host, process 400 detects whether there is a change in the voltage level at the D+ or D- line in USB bus 110. The change in the voltage level while USB device is receiving a data stream from the host indicates a change in the bit value of the incoming data stream. The detected bit may be a bit in the token packet or in any other packets following the token packet in the data stream. In response to detecting the change in the voltage level, process 400 generates a start edge, e.g., a rising edge, for a cycle of the synchronized clock signal in a step 404. Therefore, the start edge of the current cycle in the clock signal is synchronized or locked to the beginning of a bit period in the incoming packet. After generating the start edge of the synchronized clock signal in step 404, process 400 returns to step 402 with counters 106 and 108 reset to zero. Process 400 is ready for the next cycle.

No change in the voltage level indicates there is no change in the bit value. This may correspond to two situations. The first situation is that the time lapse from the previous cycle of process 400 is not equal to the time duration of one or more bits in the incoming packet because consecutive bits in the incoming packet may have the same bit value. The second situation is that USB device 100 is sending an outgoing data stream to the host. In response thereto, the counts of counters 106 and 108 increase by one a step 406. In a subsequent step 407, process 400 checks whether the count C_{106} of counter 106 satisfies Equation (1):

$$C_{106} = D \times N/8 \qquad (1)$$

In Equation (1), D is the value of the digital control signal generated in process 300 described herein above with reference to Fig. 3, and N is a positive integer.

The count C_{106} not satisfying Equation (1) indicates that the time lapse from the start edge of the synchronized clock signal is not equal to a multiple of the bit period of the incoming or outgoing data stream. In response thereto, process 400, in a step 409, checks whether the count C_{108} of counter 108 in synchronized clock signal generator 105 satisfies Equation (2):

$$C_{m} = D/16 \qquad (2)$$

The count C_{108} not satisfying Equation (2) indicates that the time lapse from the start edge of the synchronized clock signal is not equal to one half of the bit period of the data stream. In response thereto, process 400 returns to step 403 for the next cycle. If the count C_{108} satisfies Equation (2), it means that the time lapse from the start edge of the synchronized clock signal is equal to one half of the bit period of the packet. In response to such situation, process 400, in a step 412, generates a middle edge, e.g., a falling edge, for the current cycle of the synchronized clock signal. Therefore, the middle edge of a cycle in the clock signal is synchronized or locked to the midpoint of the bit period in the packet. After generating the middle edge of the synchronized clock signal in, process 400 returns to step 403 for the next cycle. In an alternative embodiment, process 400 includes an optional step of resetting the count C_{108} of counter 108 to zero after generating the middle edge for the current cycle of the synchronized clock signal in step 412 and before returning to step 403 for the next cycle.

Referring back to step 407, the count C_{106} satisfying Equation (1) indicates that the time lapse from the start edge of the synchronized clock signal is equal to a multiple of the bit period of the incoming or outgoing data stream. In response thereto, process 400, in a step 414, generates an end edge, e.g., another rising edge, for the current cycle of the synchronized clock signal. The end edge of the current cycle of the synchronized clock signal also serves as the start edge for the next cycle of the synchronized clock signal. In addition, count C_{108} of counter 108 is reset to zero in step 414. Subsequently in a step 415, process 400 verifies whether the count C_{106} satisfies Equation (3):

$$C_{106} = D \qquad (3)$$

While already satisfying Equation (1), the count C_{106} not satisfying Equation (3) indicates that the time lapse from the start edge of the synchronized clock signal is not equal to eight times of the bit period of the data stream. In response to such situation, process 400 returns to step 403 for the next cycle. If C_{106} satisfies Equation (3), the time lapse from the start edge of the synchronized clock signal is equal to eight times of the bit period of the incoming data stream. In response thereto, process 400 returns to starting step 402 and resets counters 106 and 108 to zero. After step 402, clock signal synchronization process 400 proceeds to step 403 and repeats for the next period of eight bit cycles.

It should be understood that, in accordance with the present invention, a synchronized clock signal is not limited to being generated by a process described herein above. For example, step 409 is not limited to verifying whether the count C_{108} satisfies Equation (2). In alternative embodiments, process 400, in step 409, may verify whether the count C_{106} of counter 106 satisfies Equation (4):

$$C_{106} = DxM/16 \qquad (4)$$

or Equation (5):

$$C_{106} = D \times (2M + 1)/16 \qquad (5)$$

In Equations (4) and (5), M represents an integer. In these alternative embodiments, synchronized clock signal generator 105 needs only one counter, e.g., counter 106.

In addition, process 300 described herein above with reference to Fig. 3 is not limited to setting the value D of the digital control signal to the number of periods of the reference frequency signal generated by oscillator 103 during a time duration equal to eight bit periods of the incoming token packet. The value D of the digital control signal can be set equal to the number of periods of the reference frequency signal generated by oscillator 103 during a time duration of any number of bit periods of the incoming token packet. In general, a large value D is preferred for high accuracy of synchronization. As described herein above with reference to Figs. 2 and 3, the start of the time duration is preferred not corresponding to the beginning of the first bit because it is potentially unstable. Limiting the end of the time duration so that it does not go beyond the tenth bit of the token packet is also preferred. This is because the first ten bits of the token packet is predetermined and easily identifiable in the USB protocol. Accordingly, a time duration of eight bit periods is preferred because of its

large D value, easy identification, and easy binary operation for numbers that are multiples of two, four, eight, sixteen, and so on.

The synchronized clock signal generated in process 400 described herein is locked to the data stream at element 102 of USB device 100. The synchronized clock signal enables element 102 to properly perform such functions as reading data from the host, recording and processing the data, sending data and commands to the host, etc. As pointed out herein above, USB device 100 can be a USB mouse, USB DVD player, USB MP3 player, USB rewritable optical memory, USB hard drive, USB flash memory, printer, etc. The synchronized clock signals enables element to perform a wide range of functions. It should be understood that a clock signal synchronization system or process in accordance with the present invention can be used in any digital data transmission apparatus. USB device 100 is just an example for purpose of illustration.

By now it should be appreciated that a system and a process for synchronizing or locking a clock signal to a data signal have been provided. The synchronization system in accordance with present invention can include a simple and cost efficient RC oscillator and simple digital circuitry. Such a system has the qualities of small chip size, reliable operation, and cost efficiency. The synchronization process in accordance with the present invention involves only digital operations that can be achieved in only one hand shake. Therefore, it is simple, fast, reliable, and unsusceptible to variations in chip fabrication processes and operation conditions.

While specific embodiments of the present invention have been described herein above, they are not intended as limitations on the scope of the invention. The present invention encompasses those modifications and variations of the described embodiments that are obvious to those skilled in the art. For example, although the specification describes the synchronization process in conjunction with a USB protocol for low speed signal transmission, present invention encompasses clock signal synchronization systems and processes in various data transmission protocols at various speeds.